

**APLIKASI KALENDER UNTUK MENGELOLA JADWAL KEGIATAN  
DI UMS MENGGUNAKAN *GOOGLE APPLICATION API***

Naskah Publikasi

Program Studi Informatika

Fakultas Komunikasi dan Informatika



Diajukan oleh:

*Suyadi*

*Gunawan Ariyanto, Ph.D*

**PROGRAM STUDI INFORMATIKA  
FAKULTAS KOMUNIKASI DAN INFORMATIKA  
UNIVERSITAS MUHAMMADIYAH SURAKARTA  
JUNI 2016**

## HALAMAN PENGESAHAN

Publikasi ilmiah dengan judul:

### APLIKASI KALENDER UNTUK MENGELOLA JADWAL KEGIATAN DI UMS MENGGUNAKAN *GOOGLE APPLICATION API*

Dipersiapkan dan disusun oleh:

Suyadi

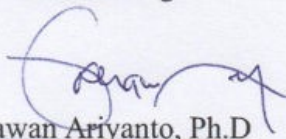
L200100015

Telah disetujui pada:

Hari : ..*Jumat*..

Tanggal : ..*5 Februari 2016*..

Pembimbing



Gunawan Ariyanto, Ph.D

NIK: 968

Publikasi ini telah diterima sebagai salah satu persyaratan

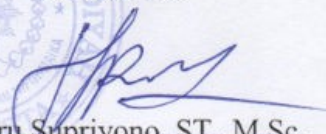
untuk memperoleh gelar sarjana

Tanggal ..*5 Feb. 2016*..

Mengetahui,

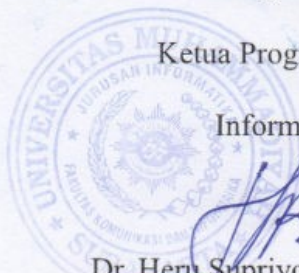
Ketua Program Studi

Informatika



Dr. Heru Supriyono, ST., M.Sc.

NIK: 907





UNIVERSITAS MUHAMMADIYAH SURAKARTA  
FAKULTAS KOMUNIKASI DAN INFORMATIKA  
PROGRAM STUDI INFORMATIKA

Jl. A. Yani Tromol Pos 1 Pabelan Kartasura Telp. (0271)717417, 719483 Fax (0271)714448  
Surakarta 57102 Indonesia. Web: <http://informatika.ums.ac.id> Email: [informatika@ums.ac.id](mailto:informatika@ums.ac.id)

---

**SURAT KETERANGAN LULUS PLAGIASI**

Assalamu'alaikum wr. wb.

Biro Skripsi Program Studi Informatika menerangkan bahwa:

Nama : SUYADI

NIM : L200100015

Judul : Aplikasi Kelender untuk Mengelola Jadwal Kegiatan di UMS  
Menggunakan *Google Application API*

Program Studi : Informatika

Status : **Lulus**

Adalah benar-benar lulus pengecekan plagiasi dari Naskah Publikasi Skripsi, dengan menggunakan aplikasi Turnitin.

Demikian keterangan ini dibuat agar dipergunakan sebagaimana mestinya.

Wassalamu'alaikum wr. wb.

Surakarta, ..... Agustus 2016..

Biro Skripsi Informatika

**Endang Wahyu Pamungkas, S.Kom., M.Kom.**

# Aplikasi Kalender untuk Mengelola Jadwal Kegiatan di UMS

## Menggunakan Google Application Api

### ORIGINALITY REPORT

0%

SIMILARITY INDEX

0%

INTERNET SOURCES

0%

PUBLICATIONS

%

STUDENT PAPERS

### PRIMARY SOURCES

EXCLUDE QUOTES ON

EXCLUDE  
BIBLIOGRAPHY ON

EXCLUDE MATCHES < 4%



Cahyo Widodo | User Info | Messages | Instructor | English



Turnitin Document Viewer - Mozilla Firefox

[https://turnitin.com/dv?lang=en\\_us&o=686056091&u=1008009566&s=3](https://turnitin.com/dv?lang=en_us&o=686056091&u=1008009566&s=3)

Copy of manajemenekonomi - DUE 09-Nov-2016RoadmapPaper 6 of 10

OriginalityGradeMarkPeerMark

Aplikasi  
BY SUYADI SUYADI



0%  
SIMILAR  
OUT OF 99

### Aplikasi Kalender untuk Mengelola Jadwal Kegiatan di UMS Menggunakan *Google Application Api*

Suyadi<sup>1</sup>, Gunawan Ariyanto<sup>1</sup>

<sup>1</sup> Jurusan Informatika, Fakultas Komunikasi dan Informatika  
Univesitas Muhammadiyah Surakarta  
(suyadi, gariyanto)@ums.ac.id

**ABSTRAKSI**

Tujuan dari penelitian ini adalah untuk membangun sebuah aplikasi untuk mengelola acara kegiatan di Universitas Muhammadiyah Surakarta (UMS) menggunakan Google Application API. Aplikasi ini dapat mempublikasikan jadwal perkuliahan sebagai pelengkap dari Aplikasi Jadwal Terpadu yang sudah ada dan menambahkan fitur untuk mengelola agenda kegiatan lainnya. Kami menggunakan framework FAST (Framework for the Application of Systems Thinking) dalam membangun aplikasi ini. FAST mendefinisikan taken untuk mengidentifikasi dan menganalisis permasalahan, kemudian

Match Overview

There are no matching sources for this report.

12

PAGE: 1 OF 11





Text-Only Report

# **APLIKASI KALENDER UNTUK MENGELOLA JADWAL KEGIATAN DI UMS MENGGUNAKAN *GOOGLE APPLICATION API***

**Suyadi<sup>1</sup>, Gunawan Ariyanto<sup>1</sup>**

<sup>1</sup> Jurusan Informatika, Fakultas Komunikasi dan Informatika  
Universitas Muhammadiyah Surakarta  
{suyadi, gariyanto}@ums.ac.id

## **ABSTRAKSI**

Tujuan dari penelitian ini adalah untuk membangun sebuah aplikasi untuk mengelola acara kegiatan di Universitas Muhammadiyah Surakarta (UMS) menggunakan Google Application API. Aplikasi ini dapat mempublikasikan jadwal perkuliahan sebagai pelengkap dari Aplikasi Jadwal Terpadu yang sudah ada dan menambahkan fitur untuk mengelola agenda kegiatan lainnya. Kami menggunakan framework FAST (Framework for the Application of Systems Thinking) dalam membangun aplikasi ini. FAST mendefinisikan tahap untuk mengidentifikasi dan mengevaluasi permasalahan, kesempatan, hambatan yang terjadi, dan kebutuhan yang diharapkan sehingga dapat mengusulkan beberapa perbaikan. Hasil pengujian kinerja menunjukkan bahwa aplikasi dapat digunakan dengan baik sesuai batasan yang ditetapkan Google. Namun, untuk mencapai kinerja terbaik membutuhkan server dengan processor spesifikasi tinggi dan/atau memanfaatkan teknologi load balancing yang memungkinkan beban server dibagi dalam beberapa server.

**Kata kunci:** Sistem Informasi, Google, Calendar, Schedule, Python, Django

## **ABSTRACT**

The aim of this research is to build an application for Managing Events in Universitas Muhammadiyah Surakarta (UMS) using Google Application API. This application can publish the classes timetable as complementary of an existing Integrated Lecture Schedule Application and add features to manage the agenda of other activities. We used FAST framework (Framework for the Application of Systems Thinking) in building this application. It defines the stages to identify and evaluate the problems, opportunities, barriers that occur, and the expected demand so it can propose some improvements. The performance testing results show that the application can normally be used correctly in accordance Google restrictions set. However, to achieve the best performance it needs a server with high specifications of a processor. It was also important to utilize load balancing technology which allows the load server divided into multiple servers.

**Keywords:** Information System, Google, Calendar, Schedule, Python, Django

## **1. PENDAHULUAN**

Pada saat ini, aplikasi kalender berbasis web sudah dipakai secara luas. Banyak perguruan tinggi yang menggunakan aplikasi kalender berbasis web untuk mengelola berbagai kegiatan, misalnya University of Minnesota menggunakan kalender untuk menjadwalkan kegiatan akademik, membuat kalender pribadi, berbagi kalender dengan orang lain dan mengundang teman ke suatu kegiatan [Skills 2014].

UMS sudah memiliki aplikasi penjadwalan yaitu aplikasi Jadwal Terpadu yang terintegrasi dengan sistem informasi akademik. Aplikasi ini dirancang untuk membantu menyusun

jadwal kuliah secara mudah dengan mengefisienkan penggunaan sumber daya dosen dan ruang sambil menghindari adanya tabrakan jadwal bagi peserta kuliah. Aplikasi ini dapat diakses melalui intranet atau jaringan di dalam kampus dan internet [UMS 2012]. Ada beberapa kekurangan dari aplikasi penjadwalan ini, diantaranya: belum ada fitur penjadwalan kegiatan non perkuliahan, tampilan jadwal masih sederhana, belum ada notifikasi ke pengguna, dan masih sering terjadi kesalahan terutama saat diakses oleh banyak orang seperti *loading* yang lama dan muncul pesan kesalahan karena gagal mengambil data dari *database*.

*Google Calendar* adalah layanan kalender yang disediakan *Google* secara gratis bagi pelanggannya. Pemilik akun *Google* dapat membuat kalender, membuat acara dan mengundang orang lain ke dalam acara tersebut. *Google Calendar* dapat diakses melalui telepon genggam dan dapat memberi peringatan melalui SMS atau Surel. *Google* menyediakan *Application API* yang memungkinkan pengembang untuk membuat aplikasi yang dapat mengakses aplikasi-aplikasi *Google* seperti *Gmail*, *Calendar* dan aplikasi lain menggunakan *Google Data APIs*, *Gadgets*, dan *Google Apps Script* [Google 2013].

Berdasarkan uraian di atas, penulis mengusulkan pembuatan aplikasi kalender berbasis web menggunakan *Google Application API* untuk mempublikasikan jadwal kuliah sebagai pelengkap aplikasi Jadwal Terpadu yang telah ada dan menambah fitur untuk mengelola agenda kegiatan lainnya, seperti: rapat, seminar, dan pameran serta fitur lain yang belum disediakan.

Dengan dibangunnya aplikasi ini diharapkan program studi dan pengampu lebih mudah dalam menginformasikan jadwal perkuliahan kepada mahasiswa, pengguna dapat mengetahui dengan mudah kegiatan yang diselenggarakan dalam suatu ruangan, dan pengguna dapat memperoleh notifikasi kegiatan yang diikutinya melalui surel dan SMS.

## **2. KEBUTUHAN SISTEM**

### **2.1. Pemrograman Python**

*Python* diciptakan oleh Guido van Rossum di Belanda pada tahun 1990 dan namanya diambil dari acara televisi kesukaan Guido *Monty Python's Flying Circus*. Van Rossum mengembangkan *Python* sebagai hobi, kemudian *Python* menjadi bahasa pemrograman yang dipakai secara luas dalam industri dan pendidikan karena sederhana, ringkas, sintaks intuitif dan memiliki pustaka yang luas [Liang 2010]. Alasan lain untuk menggunakan *Python* adalah, ada penelitian yang menunjukkan bahwa *Python* dirancang dengan baik untuk pemula, bahasa telah dirancang secara eksplisit oleh penulisnya menjadi lebih mudah untuk belajar [Schuerer et al. 2008].

*Python* adalah pemrograman berorientasi objek (OOP). Data dalam *Python* adalah sebuah objek yang dibuat dari kelas (*class*). Pemrograman berorientasi objek merupakan alat ampuh untuk membuat perangkat lunak, tiap *class* dapat digunakan kembali bahkan dapat diturunkan menjadi *class* lain sehingga tidak perlu menulis kode program berkali-kali.

*Python* saat ini dikembangkan dan dikelola oleh tim relawan yang besar dan tersedia secara gratis dari *Python Software Foundation*.

## 2.2. Django Framework

*Django* adalah *webframework open source* yang ditulis dengan bahasa perograman *Python* yang dikembangkan pertama kali pada tahun 2003 oleh *programmer* koran *Lawrence Journal-World* Adrian Holovaty dan Simon Willison. Tahun 2005 *Django* dirilis sebagai proyek *open source*, hingga saat ini *Django* dikembangkan dan dimaintain oleh *Django Software Foundation* di bawah lisensi *BSD license*.

Sejak *Django* dirilis sebagai proyek *open source* hingga saat ini terdapat ribuan perusahaan dan organisasi di dunia yang menggunakannya dalam proyek besar maupun kecil, di antaranya: *The Washington Post*, *The Lawrence Journal-World*, *Google*, *EveryBlock*, *Newsvine*, *Curse Gaming*, *Tabblo* dan *Pownce* [Forcier et al. 2009].

*Django* menyediakan *high-level framework* yang dapat digunakan untuk membangun aplikasi *web* dengan sedikit baris kode, simpel, kuat, fleksibel dan mudah.

## 2.3. JavaScript, JSON, AJAX dan jQuery

*JavaScript*, diperkenalkan sebagai *LiveScript* oleh *Netscape Communications Corp*, telah berkembang pesat dalam beberapa tahun terakhir. *JavaScript* pada awalnya dikembangkan untuk membuat halaman *web* lebih interaktif, dan mengontrol perilaku halaman. Program *JavaScript* biasanya tertanam di dalam file *HTML*. *HTML* adalah bahasa markup, dan tidak memanipulasi perilaku halaman sekali yang dimuat. Menggunakan *JavaScript*, pengembang web dapat menetapkan aturan dan memverifikasi jika aturan itu diikuti, menghindari sumber daya *remote server* untuk validasi input atau bilangan kompleks angka. Hari ini *JavaScript* tidak hanya digunakan untuk validasi input dasar; digunakan untuk mengakses objek Dokumen *browser*, untuk membuat panggilan *asynchronous* ke *server web*, dan untuk mengembangkan aplikasi *web end-to-end* dengan menggunakan *platform* perangkat lunak seperti *Node.js*, yang didukung oleh *v8 Google JavaScript engine* [Sriparasa 2013, h.5].

*JSON (JavaScript Object Notation)* adalah format pertukaran data yang sangat populer yang dikembangkan oleh Douglas Crockford. *JSON* berbasis teks, ringan dan mudah dibaca manusia untuk pertukaran data antara klien dan server. *JSON* berasal dari *JavaScript* dan memiliki kemiripan dengan objek *JavaScript*, tetapi tidak tergantung *JavaScript*. *JSON* adalah bahasa-independen, dan dukungan untuk format *JSON* tersedia dalam semua bahasa pemrograman populer seperti: *C#*, *PHP*, *Java*, *C++*, *Python*, dan *Ruby* [Sriparasa 2013, h.15].

Permintaan *web* dapat dilakukan dengan dua cara, yaitu sinkron dan asinkron. Dalam permintaan sinkron segala sesuatu harus dilakukan secara teratur, di mana langkah berikut harus menunggu sampai yang sebelumnya telah menyelesaikan eksekusi. Misalnya ada empat komponen independen pada halaman *web* saat halaman dimuat; jika salah satu komponen membutuhkan waktu yang lama selama eksekusi, sisa halaman akan menunggu sampai eksekusi komponen tersebut selesai. Jika eksekusi gagal, pemanggilan halaman gagal juga. Untuk mengatasi masalah pada permintaan sinkron, dikembangkan permintaan asinkron. Produk pertama ada-



lah tag *IFrame* yang dikenalkan oleh *Microsoft*. *Jesse James Garrett* memperkenalkan produk yang cukup fenomenal *AJAX*. *AJAX* (*Asynchronous JavaScript and XML*) adalah permintaan *web* yang dilakukan melalui *JavaScript* dan pertukaran data yang pada awalnya menggunakan format *XML*. "X" di *AJAX* merujuk pada penggunaan *XML* dalam pertukaran data, tetapi sekarang dapat pula menggunakan *JSON*, file teks, bahkan *HTML* [Sriparasa 2013, h.44].

## **2.4. Google Apps Platform**

*Google Apps Platform* adalah sejumlah besar aplikasi dan layanan yang terintegrasi satu sama lain yang dikelompokkan dalam tiga kategori. Pertama, alat komunikasi yang terdiri dari *Gmail*, sebuah *email* sistem yang sekarang banyak dipakai secara luas, *Google Calendar* yaitu agenda *online* dengan fasilitas berbaginya, dan *Google Talk* yang digunakan untuk mengobrol secara *online*. Kedua, perangkat kolaborasi yang terdiri dari *Google Sites*, *Google Docs* dan *Address Book*. Terakhir, perangkat keamanan yang dikenal dengan *Postini Services*. Semua aplikasi dan *service Google Apps* dikaitkan dengan sebuah *domain* [Google 2011].

## **2.5. Google Application API**

*Google Application API* memungkinkan pengembang aplikasi untuk menulis aplikasi yang mengakses aplikasi *Google* seperti *Gmail*, *Calendar* dan lainnya menggunakan *Google Data API*, *Gadget*, dan *Google Apps Script* [Google 2013].

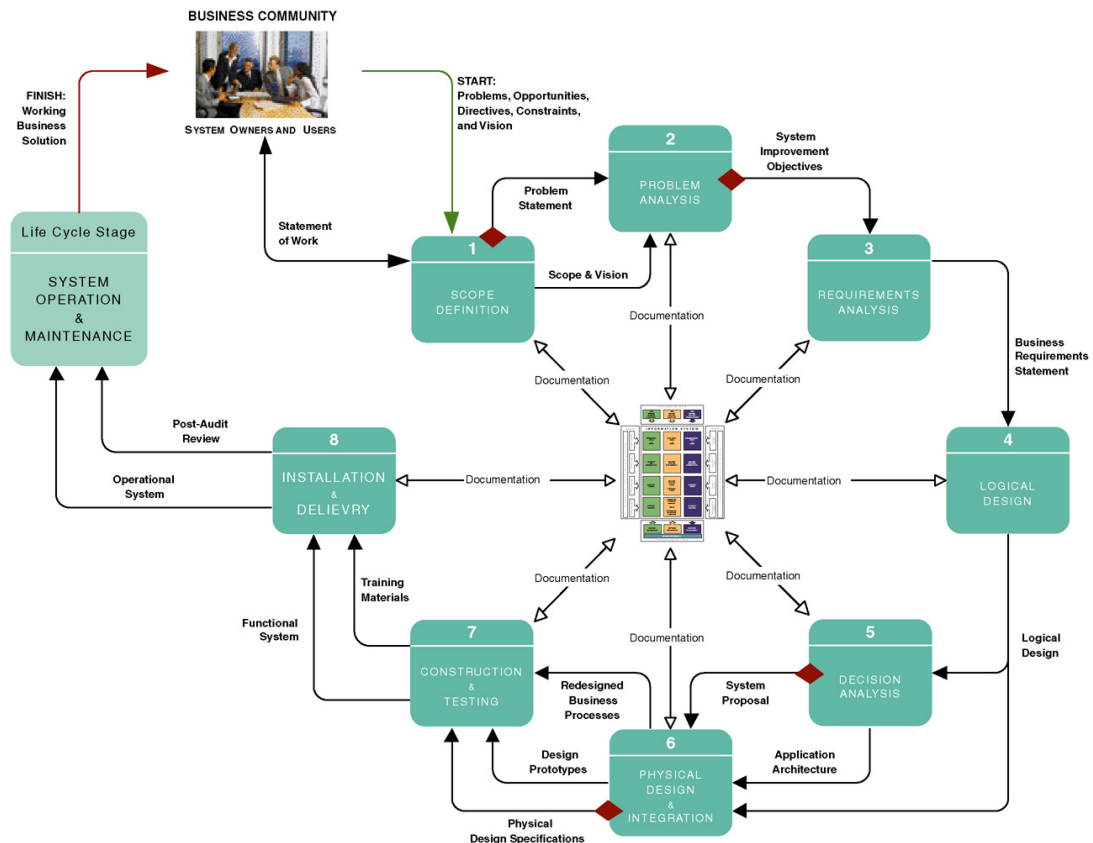
*Google* menyediakan protokol sederhana *Google Data* yang dapat digunakan para pemrogram untuk membuat aplikasi yang membaca dan menulis data di *internet* yang disediakan oleh *Google*. Beberapa layanan *Google* menyediakan akses data eksternal melalui *API* yang memanfaatkan protokol ini [Google 2012]. Protokol tersebut adalah *AtomPub* dan *JSON*.

*Google Apps API* yang digunakan penulis dalam penelitian ini adalah *Google Calendar Resource API* dan *Google Calendar API*. *Google Resource API* digunakan untuk mengelola *data resource* seperti: ruang, kendaraan dan komputer. *Google Calendar API* digunakan untuk mengelola data *Calendar*.

## **3. METODE PENGEMBANGAN**

Dalam mengembangkan aplikasi ini penulis menggunakan *framework FAST* (*Framework for the Application of Systems Thinking*). *FAST* mendefinisikan tahapan untuk mengidentifikasi dan mengevaluasi permasalahan, kesempatan, hambatan yang terjadi, dan kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikan.

Gambar 1 mengilustrasikan tahapan metodologi *FAST*. Setiap tahapan menghasilkan satu dokumen yang dibutuhkan pada tahapan berikutnya. Proyek dimulai dengan beberapa kombinasi dari *Problems*, *Opportunities*, *Directives*, *Constraints and Vision* dari pengguna (panah hijau) dan diakhiri dengan *Working Business Solution* (panah merah) bagi pengguna [Whitten and Bentley 2007, h.80].



**Gambar 1. Proses Pengembangan Sistem**

**Scope Definition.** Meliputi identifikasi masalah, mendefinisikan visi, ruang lingkup, batasan, kebutuhan pengguna, jadwal, dan anggaran.

**Problem Analysis.** Digunakan untuk mempelajari sistem yang sudah ada dan menganalisis temuan supaya tim proyek memiliki pemahaman yang menyeluruh tentang masalah yang akan dikerjakan dalam proyek tersebut.

**Requirement Analysis.** Menjawab pertanyaan: kemampuan apa yang harus disediakan sistem baru bagi penggunanya? data apa yang harus ditangkap dan disimpan? apa tingkat kinerja yang diharapkan? apa prioritas dari berbagai kebutuhan?

**Logical Design.** Terjemahan dari kebutuhan pengguna bisnis ke dalam suatu model sistem yang menggambarkan hanya kebutuhan bisnis dan tidak ada desain teknis pelaksanaan kebutuhan tersebut.

**Decision Analysis.** Mengevaluasi kandidat sistem ditinjau dari kelayakan teknis, operasional, ekonomi, resiko dan waktu yang dibutuhkan dalam pengembangan.

**Physical Design and Integration.** Terjemahan dari kebutuhan pengguna bisnis ke dalam suatu model sistem yang menggambarkan implementasi teknis kebutuhan bisnis pengguna.

**Construction and Testing.** Fase pembangunan sistem yang meliputi: perangkat lunak, database, antarmuka sistem dan pengguna, perangkat keras, dan perangkat jaringan.

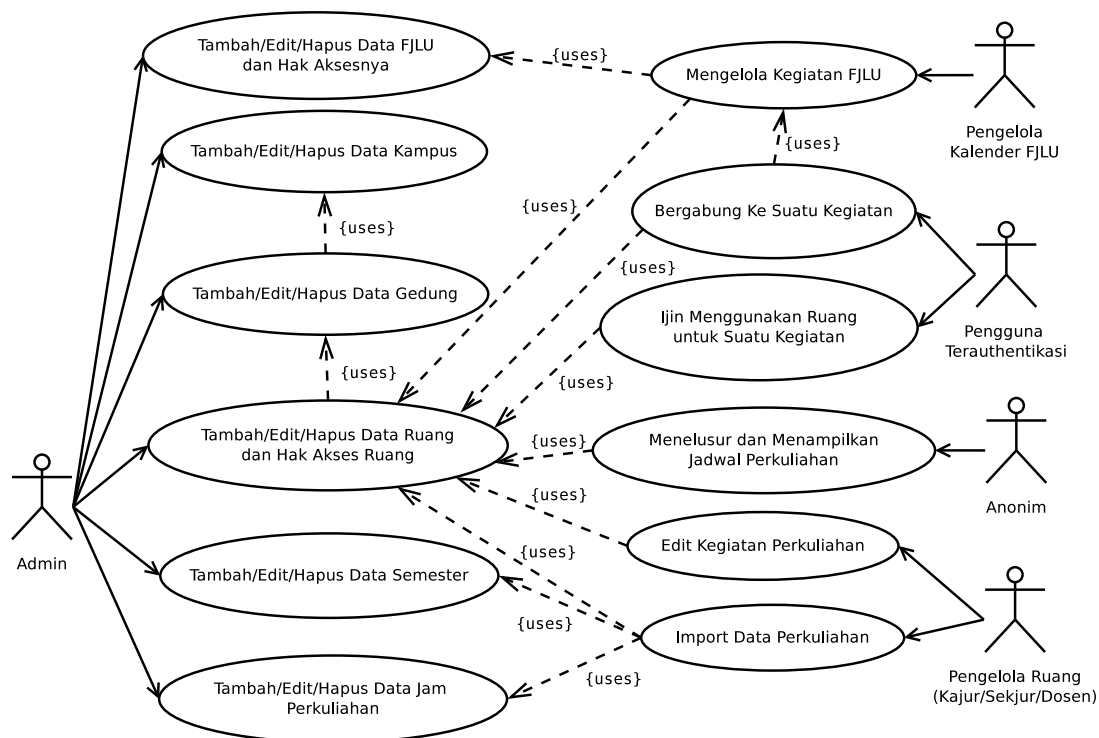
**Installation and Delivery.** Fase instalasi dan penyerahan mencakup: instalasi sistem untuk dio-  
perasikan yang sesungguhnya (production), pelatihan pengguna, melengkapi dokumentasi, dan  
konversi data yang ada.

## 4. HASIL PENGEMBANGAN

### 4.1. Deskripsi Sistem

Aplikasi Kalender untuk mengelola jadwal kegiatan di UMS menggunakan *Google Application Api* dibangun untuk mempublikasikan jadwal kuliah sebagai pelengkap aplikasi Jadwal Terpadu yang telah ada dan menambah fitur untuk mengelola agenda kegiatan lainnya.

Kelebihan dari aplikasi kalender ini adalah integrasi dengan *Google Calendar*. Semua ruangan direpresentasikan sebagai *Google Calendar Resource* yang masing-masing *resource* memiliki sebuah *calendar sendiri* dan masing-masing *calendar* memiliki *event*. Tiap *event* me-representasikan suatu kegiatan, misalnya: kegiatan perkuliahan. Data *Google Calendar* dapat diakses menggunakan aplikasi kalender dalam telpon cerdas dan *Google* juga dapat memberi notifikasi melalui *email* dan *SMS*.



**Gambar 2. Desain Aplikasi**

Gambar 2 adalah diagram desain aplikasi yang dikembangkan yang menggambarkan aktor (pengguna aplikasi), proses-proses dalam aplikasi dan hubungan antar proses.

Aktor dari aplikasi ini terdiri dari administrator yang memiliki hak penuh, pengelola ruangan yang memiliki hak untuk mengelola jadwal kegiatan di dalam ruang tersebut, pengelola kalender FJLU (Fakultas, Jurusan, Lembaga dan UNIT di UMS) yang berhak untuk mengelola

kegiatan FJLU, pengguna terautentikasi yaitu pengguna yang berhasil login ke dalam aplikasi ini (mahasiswa, dosen, dan karyawan) yang memiliki hak untuk memesan suatu ruangan atau bergabung dalam suatu kegiatan, dan pengguna anonim yaitu orang yang mengakses aplikasi tanpa autentikasi.

#### 4.1.1. Tampilan untuk Pengguna

Pengguna mengakses program menggunakan *web browser* dengan membuka alamat *URL* dari *server*. Halaman yang pertama kali muncul adalah tampilan halaman depan yang memuat menu-menu dan daftar kampus. Ketika pengguna memilih kampus tertentu akan ditampilkan daftar gedung dan peta kampus tersebut. ketika pengguna memilih gedung tertentu akan ditampilkan daftar ruang dalam gedung tersebut beserta kalendernya. Kalender dapat digeser ke depan dan ke belakang, dapat ditampilkan dalam format bulanan, mingguan dan harian. Kegiatan dalam kalender dapat ditampilkan detailnya dengan mengklik nama kegiatan. Pengguna harus melakukan login untuk menggunakan fasilitas tertentu yang disediakan oleh program seperti: membuat kegiatan, mengubah data kegiatan, mendaftar ke sebuah kegiatan, dan lain-lain. Pengguna dengan hak administrator dapat mengelola data kampus, gedung, ruang dan memberi hak pengelolaan kalender ruangan kepada pengguna lain.

[Home](#) > [Kampus 2](#) > Gedung J

<

>

hari ini

Desember 2015

Bulan

Minggu

Hari

☒ Semua

Sen	Sel	Rab	Kam	Jum	Sab	Min	
30	1	2	3	4	5	6	
7 Pemrograman	7 Computer	7 Algoritma	7 Pemrograman	7 Logika Kabin	9:30 Interaksi		<input checked="" type="checkbox"/> J04011
8:40 Etika & I	7 Kalkulus /	7 Etika & Hul	7 Pancasila /	7 Sistem Ope	12:30 Admin		<input checked="" type="checkbox"/> J0402
9:30 Manajer	7 Teknologi I	8:40 Etika &	8:40 Techno	9:30 Sistem I	12:30 Organi		<input checked="" type="checkbox"/> J0403
12:30 Manajer	9:30 Metode	10:20 Etika &	9:30 English	9:30 Techno			<input checked="" type="checkbox"/> J0405
12:30 Etika &	9:30 Probabi	10:20 Interni	10:20 Techni	13 Sistem M			<input checked="" type="checkbox"/> J0406
	9:30 Teknolo	12:30 Etika &	12:30 Metod	13 Technopr			<input checked="" type="checkbox"/> J0407
	12:30 Metod	12:30 Manajer	12:30 Teknik				<input checked="" type="checkbox"/> J0408
	12:30 Teknol						<input checked="" type="checkbox"/> J0409
	12:30 Keper						<input checked="" type="checkbox"/> JSEM1
7	8	9	10	11	12	13	<input checked="" type="checkbox"/> JSEM2
7 Pemrograman	7 Computer	7 Algoritma	7 Pemrograman	7 Logika Kabin	9:30 Interaksi		<input checked="" type="checkbox"/> RuangRapatFKI
8:40 Etika & I	7 Kalkulus /	7 Etika & Hul	7 Pancasila /	7 Sistem Ope	12:30 Admin		
9:30 Manajer	7 Teknologi I	8:40 Etika &	8:40 Techno	9:30 Sistem I	12:30 Organi		
12:30 Manajer	9:30 Metode	10:20 Etika &	9:30 English	9:30 Techno			
12:30 Etika &	9:30 Probabi	10:20 Interni	10:20 Techni	13 Sistem M			
	9:30 Teknolo	12:30 Etika &	12:30 Metod	13 Technopr			
	12:30 Metod	12:30 Manajer	12:30 Teknik				
	12:30 Teknol						

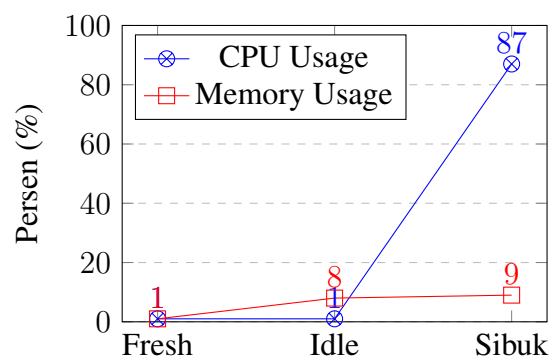
Gambar 3. Kalender Kegiatan dalam Gedung

#### 4.1.2. Administrasi

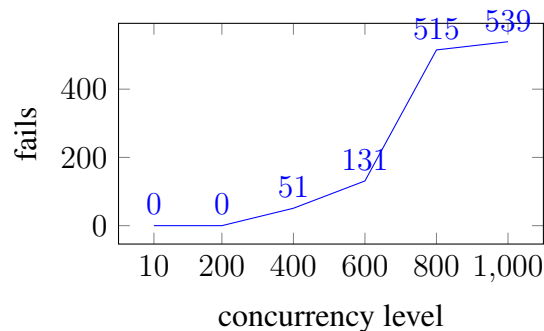
Tampilan administrasi digunakan untuk mengelola (menambah, mengubah dan menghapus) data kampus, gedung, ruang dan data-data lain yang dibutuhkan dalam aplikasi, misalnya mengatur jam perkuliahan. Tampilan administrasi juga digunakan untuk mengatur hak akses pengguna ke sistem.

#### 4.2. Pengujian

Penulis melakukan tiga buah pengujian pada aplikasi yang dibuat, yaitu: menguji kinerja server, menguji kinerja web server dan menguji beberapa fungsional aplikasi. Hasil pengujian tampak seperti gambar 4 dan gambar 5.



Gambar 4. Statistik Pemakaian *Processor* dan Memori Server



Gambar 5. Hasil Pengujian Menggunakan *ApacheBench* sejumlah 10000 request dengan level konkuren 10, 100, 200, 400, 600, 800 dan 1000

Pengujian kinerja server digunakan untuk mengetahui pemakaian *processor* dan memori. Dari pengujian ini tampak bahwa *processor* bekerja keras jika permintaan ke server banyak, sementara memori hanya terpengaruh sedikit.

Dari pengujian kinerja *web server* tampak bahwa web server optimal pada konkurensi request di bawah 600. Untuk konkurensi di atas 600 banyak terjadi *fails*.

Penulis melakukan akses menggunakan tablet *android* dan mencatat beberapa notifikasi melalui surel dan SMS untuk menguji fungsionalitas sistem.

### 4.3. Pembahasan

Aplikasi kalender untuk mengelola jadwal kegiatan di UMS menggunakan *Google Application API* dikembangkan sebagai alternatif dan tambahan Aplikasi Jadwal Terpadu dalam menginformasikan kegiatan perkuliahan kepada pengguna. Kelebihan dari aplikasi ini adalah adanya informasi grafis berupa peta untuk memudahkan pengguna mencari lokasi tertentu. Aplikasi ini diintegrasikan dengan *Google Calendar* yang memiliki fitur yang cukup lengkap di antaranya: notifikasi melalui email dan SMS dan dapat diakses menggunakan telpon cerdas atau tablet.

*Framework Fast* yang digunakan penulis untuk merancang aplikasi ini sangat membantu dalam mencapai tujuan mulai dari tahap perancangan hingga tahap akhir. Semua proses terkontrol, kesalahan-kesalahan yang terjadi pada semua tahapan dapat diperbaiki tanpa harus membuat rancangan mulai dari awal.

Berdasarkan pengujian, aplikasi ini pada umumnya dapat digunakan dengan baik sesuai batasan yang ditetapkan *Google*, tetapi untuk kinerja yang baik perlu server dengan *processor* spesifikasi tinggi dan/atau memanfaatkan teknologi *load balancing* yang memungkinkan beban server dibagi dalam beberapa server.

## 5. PENUTUP

### 5.1. Simpulan

Berdasarkan uraian dari pembangunan sistem dan pengujian sistem, dapat diperoleh kesimpulan sebagai berikut:

**Pertama**, aplikasi kalender untuk mengelola jadwal kegiatan di UMS menggunakan *Google Calendar API* telah berhasil dibangun.

**Kedua**, aplikasi kalender untuk mengelola jadwal kegiatan di UMS menggunakan *Google Calendar API* dapat membantu *civitas akademika* dalam menginformasikan kegiatan perkuliahan dan non perkuliahan dalam bentuk *calendar event*.

**Ketiga**, pengguna sistem ini dapat mengakses data kalendernya menggunakan perangkat telpon cerdas atau tablet dan menerima pemberitahuan melalui *email* dan SMS.

**Keempat**, hak akses ke dalam kalender di sistem ini sesuai dengan hak akses dari *Google Calendar* yang dapat diatur oleh administrator sistem.

**Kelima**, kelemahan dari sistem ini adalah penggunaan *processor* yang tinggi dan batasan *query* yang hanya 1.000.000 query per hari yang dirasa kurang untuk kebutuhan seluruh UMS.

## PERSANTUNAN

Penelitian ini berhasil dilaksanakan berkat adanya program beasiswa studi lanjut Pengembangan Sumber Daya Manusia bagi tenaga kependidikan Universitas Muhammadiyah Surakarta.



## DAFTAR PUSTAKA

- Forcier, J., Bissex, P., and Chun, W. (2009). *Python Web Development with Django*. Pearson Education, Inc, Boston.
- Google (2011). *Google Apps : Mastering Integration and Customization*. Packt Publishing Ltd.
- Google (2012). Google data apis protocol reference (online). Url: [https:// developers.google.com/gdata/docs/2.0/reference](https://developers.google.com/gdata/docs/2.0/reference). diakses 19 Des. 2014.
- Google (2013). Google apps application apiss (online). diakses 6 Okt. 2013.
- Liang, Y. D. (2010). *Introduction to Programming using Python*. Boston: Pearson.
- Schuerer, K., Maufrais, C., Letondal, C., Deveaud, E., and Petit, M.-A. (2008). *Introduction to Programming using Python: Programming Course for Biologists at the Pasteur Institute*. Pasteur Institute.
- Skills, I. (2014). Using google (online). Url: <http://it.umn.edu/using-google-calendar-0>. diakses 21 Des. 2014.
- Sriparasa, S. S. (2013). *JavaScript and JSON Essentials*. Packt Publishing.
- UMS (2012). Jadwal terpadu (online). Url: <https://ajt.ums.ac.id/>. diakses 6 Okt. 2013.
- Whitten, J. L. and Bentley, L. D. (2007). *Systems Analysis and Design Methods*. McGraw-Hill/Irwin, 7th ed. edition.